REMARKS

Claims 1-38 were originally presented in the subject application. Claims 1, 9, 13, 21, 25-27 and 35 were amended in an Amendment and Response to Office Action dated November 14, 2005. Claims 1, 9, 13, 21, 25, 26, 27 and 35 have herein been amended, and claims 3 and 15 canceled without prejudice, to more particularly point out and distinctly claim the subject invention. No claims have herein been added or canceled. Therefore, claims 1, 2, 4-14 and 16-38 remain in this case.

The addition of new subject matter has been scrupulously avoided. In that regard, support for the common amendments to the independent claims can be found throughout the specification, for example, in FIG. 1A and the description thereof.

Applicants respectfully request entry of these remarks, and reconsideration and withdrawal of the sole ground of rejection.

35 U.S.C. §103 Rejection

The final Office Action rejected claims 1-38 under 35 U.S.C. §103, as allegedly obvious over Davidson et al. (U.S. Patent No. 6,042,614), hereinafter "Davidson," in view of Li et al. (U.S. Patent Application Publication No. 2003/0056200), hereinafter "Li." Applicants respectfully, but most strenuously, traverse this rejection as it applies to the amended claims.

Amended claim [illegible] recites a method of facilitating debugging of transactions. The method comprises executing a transaction on one processor of a plurality of processors, the transaction having debug information attached to the transaction. The method further comprises requesting, by the transaction, a service on another processor of the plurality of processors. The attached debug information is passed with the transaction from the one processor to the another processor, eliminating a need for attaching the debug information at the another processor. The path of the transaction is not predefined to a controller of the debugging, and at least a portion of the debug information is used to automatically establish a new debug session at the another processor without intervention. The method further comprises providing, by the controller to the one processor, at least a part of the debug information, wherein the controller is different from the one processor and the another processor, and the debug information is provided to the another processor independent of the controller.

Amended claim 1 now includes the subject matter of prior claim 3, along with a recitation that the controller is different from the one processor and the another processor.

Applicants continue to submit that neither reference, nor their combination, teaches or suggests, for example, passing information used to automatically establish a new debug session at another processor without intervention, as claimed. However, even ignoring this aspect, Applicants submit that the cited references, alone or in combination, fail to disclose, teach or suggest, for example, providing the debug information to another processor independent of the controller of the debugging.

Wile Davidson may teach a debug session generally, the debug session is not being set up to a second processor without intervention, as claimed. There may be some confusion with the term "client host" at page 4 of the final Office Action. Use of the term in Davidson at column 2, line 30 refers to the first processor that acts like a client when calling a service on the second processor. Regardless of what happens when the first processor connects to the second in Davidson in a situation without debugging, it is clear that if debugging is taking place, intervention is required to ensure that the debugger can connect to the second processor. In this case, FIG. 10 of Davidson and the cited description thereof make clear that a determination as to whether a dbx engine is running on the remote host is communicated to the client host by the server. Thus, nothing happens regarding a dbx engine on the remote host in Davidson without the knowledge of the client host. The noted section of Davidson makes clear that when there is no dbx engine running on the remote host, it is the client dbx engine that initiates a request for one to be created. Thus, Applicants submit that nothing is passed to the remote host dbx engine independent of the client host, and, therefore, the noted section of Davidson does not read on the claim 1 aspect of providing the debug information to another processor independent of the controller of the debugging.

Against prior claim 3, the final Office Action also cited to Davidson at FIG. 10 and column 14, lines 50-64. However the client host is the controller in Davidson, and FIG. 10 and the cited description thereof make clear that a determination as to whether a dbx engine is running on the remote host is communicated to the client host by the server. Thus, nothing happens regarding a dbx engine on the Davidson remote host without the knowledge of the client host. Moreover, the cited section of Davidson makes clear that when there is no dbx engine running on the remote host, it is the client dbx engine that initiates a request for one to be created. Thus, Applicants submit that nothing is passed to the remote host dbx engine independent of the client host. Therefore, Davidson cannot read on the aspect of claim 1 that the debug information is provided to the another processor independent of the controller (which is different).

In addition, Applicants submit that Li fails to remedy the above-noted shortcoming of Davidson regarding the noted aspect of claim 1. The Global Causal Identifier of Li is passed between the stub and the skeleton. See Li at numbered paragraph 0059. Indeed, the Global Causal Identifier is merely logged and analyzed in a separate, later process. See Li at FIG. 8 and numbered paragraph 0144. Thus, Li does not teach passing debug information as claimed in claim 1, and used to establish a new debug session, but merely logs data for post-runtime analysis.

The Advisory Action also appeared commented on claim 3 being unclear as to whether the controller comprised either of the processors recited. The amendment herein makes clear that the controller of the debugging is different from the other processors in the claim.

Based on the above, Applicants submit that neither reference, nor their combination, teaches or suggests passing information used to automatically establish a new debug session at another processor without intervention, as claimed in claim 1.

Therefore, Applicant submits that claim 1 cannot be rendered obvious over Davidson in view of Li.

Independent claims 9, 13, 21, 25-27 and 35 each include a limitation similar to that argued above with respect to claim 1. Thus, the remarks made with respect to claim 1 are equally applicable thereto. Therefore, claims 9, 13, 21, 25-27 and 35 also cannot be rendered obvious over Davidson in view of Li.

CONCLUSION

Applicants submit that the dependent claims are allowable for the same reasons as the independent claims from which they directly or ultimately depend, as well as for their additional limitations.

For all the above reasons, Applicants maintain that the claims of the subject application define patentable subject matter and earnestly request allowance of claims 1, 2, 4-14 and 16-38.

If a telephone conference would be of assistance in advancing prosecution of the subject application, Applicants' undersigned attorney invites the Examiner to telephone him at the number provided.

Respectfully submitted,

Wayne F. Reinke
Attorney for Applicants
Registration No.: 36,650

Dated: October 6, 2006.

HESLIN ROTHENBERG FARLEY & MESITI P.C.
5 Columbia Circle
Albany, New York 12203-5160
Telephone: (518) 452-5600
Facsimile: (518) 452-5579